

RenderManía

Número 7

Biblioteca 3D

Edición de
puntos con
Imagine 3.0

Informe

¿Quién es
Tomwoof?

Foro del lector

Elementos
para la
próxima
batalla

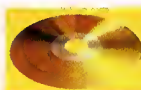
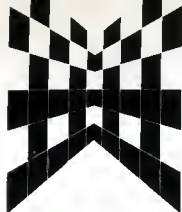
En el CD-ROM

Ray13
un trazador de
rayos español

LA PORTADA

POV + Imagine
Lo mejor de
dos mundos





Todo los ficheros de ejemplo
podéis encontrarlos en el
directorio PCMANIA\RENDER57
del CD-Rom.



José Manuel Muñoz

¿Qué fue de Tomwoof?

Hace casi dos años publicamos un artículo sobre las “mangababes” de Tomwoof. A pesar del tiempo transcurrido seguimos recibiendo modelos 3D inspirados en la obra de Tomwoof. Por lo visto, aquellas chicas 3D de estilo manga causaron una gran impresión a muchos lectores. Pero, ¿qué fue de las “mangababes”? ¿Y qué fue de Tomwoof?



La primera vez que vi el trabajo de Tomwoof fue en el CD de Walnut Creek dedicado a POV. La calidad y el encanto de estas chicas virtuales realizadas con Image 3.0 me impresionaron de tal forma que decidí dedicarles el artículo de Rendermanía incluido en el número 35, a pesar de la poca información que tenía sobre las “mangababes” o su autor. Hoy esta situación ha mejorado gracias a las páginas web publicadas sobre este particular.

Después de realizar una búsqueda con las palabras “Tomwoof” y “mangababes”, obtuvimos una serie de enlaces (en la época del primer artículo Tomwoof no

debía estar conectado a la red). Estos enlaces nos permitieron actualizar un poco nuestros datos sobre el autor y su obra. Las direcciones más interesantes son:

www.dormnet.his.se/~a95amckh/

links/manga.html

www.3dcafe.com

www.webcom.com/tomwoof.html

En la página de Tomwoof encontramos algunos datos sobre este autor cuyo nombre real es Pang Sic Piau; un “humano” de raza china, residente en Singapur. Pang es un informático que se licenció hace algunos años en la Nanyang Technological University y entre sus habilidades están la programación en C, el diseño de páginas web, el diseño en 2D y, por supuesto, el modelado en 3D. En-



tre los datos suministrados por este artista están su teléfono (65 2735363) y su e-mail (siepiiau@pacific.net.sg).

Una librería de chicas 3D

Las “mangababes” son una librería de chicas 3D que Tomwoof realizó con Imagine sin emplear ningún tipo de Ipas. Son modelos articulados que Tomwoof vende en dos modalidades en los formatos TDD (algunos llaman así al formato empleado por Imagine), 3DS, DXF y, recientemente, en el formato de Lightwave. Como podéis ver por las fotos, las chicas de Tomwoof no son modelos de estilo “real”, pero muchos opinamos que su encanto supera la fría perfección de otros modelos femeninos de librerías mucho más caras. Por otro lado, y partiendo de estos modelos, Tomwoof ha creado versiones 3D de Chunli, Cammy y otros personajes famosos de videojuegos.

Hoy publicamos la versión de demo de esta librería. En ella encontraréis una chica en los formatos de 3D Studio e Imagine, algunas imágenes y texturas de prueba, y un fichero para el registro. Según Tomwoof, la última versión de las “mangababes” (la 1.1) incluye mapas de texturas para crear diversas expresiones faciales, mapas de texturas para secuencias animadas de las caras, modelos articulables en diferentes poses, imágenes de ejemplo e instrucciones para realizar los renders.

Algunas puntualizaciones

Para saber algunos detalles acerca de cómo se pasaron los modelos de Imagine al formato 3DS, veamos qué decía el propio autor en el fichero 3DS.TXT que podéis hallar en la demo:

“La conversión desde el formato TDD del Imagine de Impulse al 3DS del 3D Studio de Autodesk es tediosa y consume mucho tiempo. Primero, los objetos son tomados y grabados uno a uno separadamente desde Imagine como objetos DXF. Después cada uno de estos objetos es importado dentro de 3D Studio y se le asignan los atributos, materiales, texturas y coordenadas de mapeado correspondientes.

Yo no suelo usar PC y no soy un experto en el uso de 3D Studio. Las versiones 3DS de las “mangababes” han sido posibles gracias a la ayuda de mis amigos de Aura Media, una firma multimedia y DTP de Singapur”

El modelo que hallaréis en la demo ha sido grabado como un único objeto. No es, por tanto, articulable. Además faltan algunas piezas vitales, como las manos, los pies y los senos. Con todo la malla nos servirá -junto con las fotos- para comprobar la calidad del trabajo de Tomwoof.

Ahora, por si alguien desea registrarse, aquí van un par de notas:

- La dirección dada por Tomwoof en el fichero ORDER.TXT incluido en la demo no es idéntica a la que hemos ha-

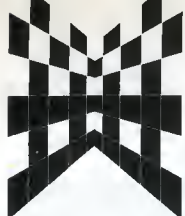
llado en uno de los ficheros HTML más recientes. Por ello recomendamos contactar mediante correo electrónico con este autor antes de proceder al registro.

- Olvidaos de lo que dice la demo sobre las opciones de registro. Parece ser que la única opción de registro disponible actualmente para la versión 1.1 es la B. De nuevo aquí parece conveniente consultar con el autor.

- Tomwoof advierte que es conveniente especificar el pago en dólares de Singapur -por problemas bancarios- para acelerar el envío. De nuevo recomendamos contactar por e-mail.

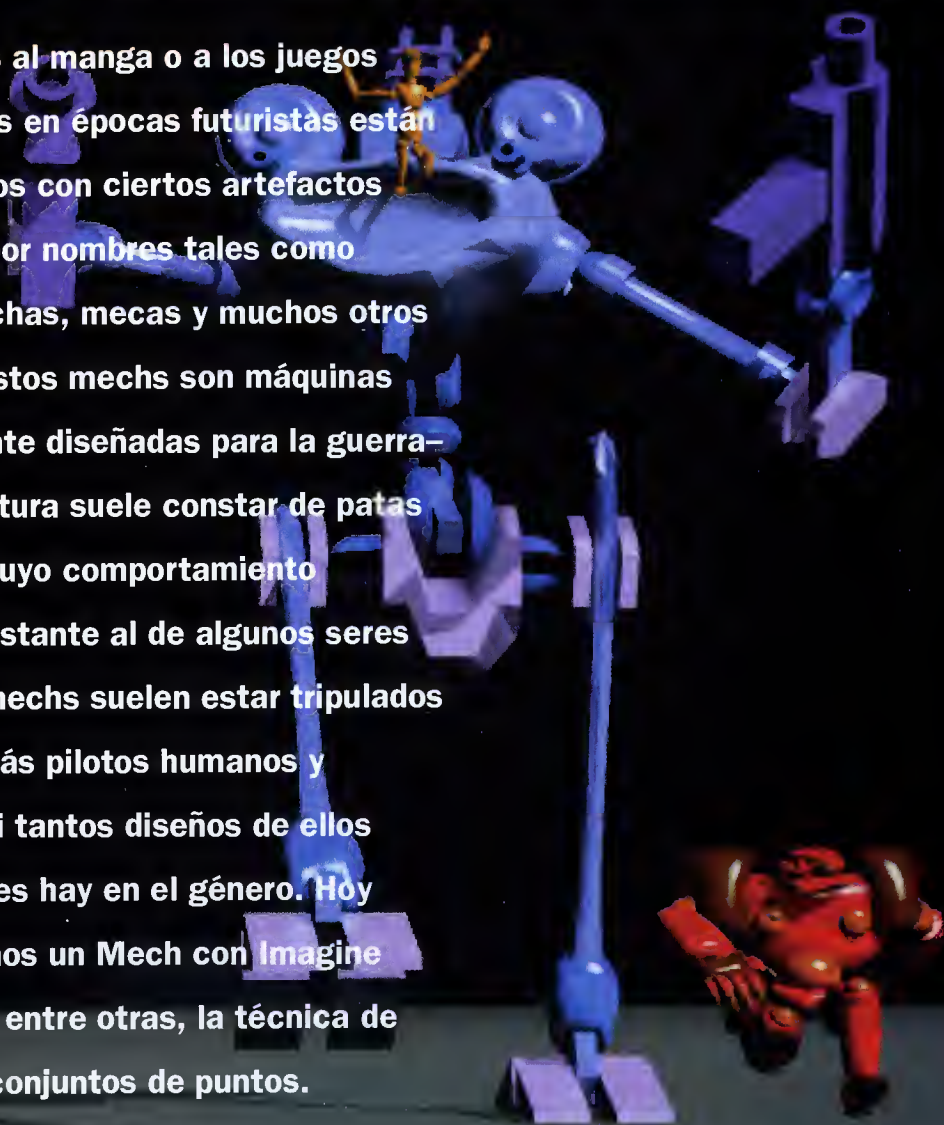
Más datos útiles

Según Tomwoof ya se está preparando la versión 2.0 de las “mangababes”. A la pregunta de cuándo estará concluida, este autor responde sencillamente: “cuando esté acabada”. Finalmente añadiremos para los curiosos que, en el web, los trabajos de Tomwoof figuran bajo el sello Wireframe, el cual es seguramente de creación propia. Otro trabajo interesante de este autor es “Virtual Warriors Design Kit”, que parece ser una librería de superhéroes de la serie Bola de Dragón. Esto es, por ahora, todo lo que sabemos de este interesante autor. Bueno, también conocemos su opinión respecto a la pregunta última acerca de la vida, el universo y todo lo demás: “leed la guía del autostopista galáctico de Adams”.



Battlemechs a medida

Los adeptos al manga o a los juegos ambientados en épocas futuristas están familiarizados con ciertos artefactos conocidos por nombres tales como Mechs, mechas, mecas y muchos otros términos. Estos mechs son máquinas –normalmente diseñadas para la guerra– cuya estructura suele constar de patas y brazos y cuyo comportamiento recuerda bastante al de algunos seres vivos. Los mechs suelen estar tripulados por uno o más pilotos humanos y existen casi tantos diseños de ellos como autores hay en el género. Hoy construiremos un Mech con Imagine empleando, entre otras, la técnica de edición de conjuntos de puntos.



El género de los mechas cuenta con infinitud de autores que, perfectos conocedores del atractivo que tienen estas máquinas entre sus muchos aficionados, se devanan los sesos intentando crear diseños atractivos, algunos de los cuales están basados en ideas muy detalladas y originales.

En Appleseed, por ejemplo, Shirow nos muestra unos Mechs que son poco más que armaduras de gran tamaño tripuladas por seres humanos. En estas armaduras, las extremidades del mech “copian” los movimientos de los miembros del piloto humano. (Shirow ha diseñado un ingenioso sistema de articulaciones). En otras obras el sistema de control es diferente. Nuestro Mech de hoy está basado en las impresionantes máquinas del juego “Battle-tech” (en el que se supone que el piloto emplea un cybergasco para controlar al mech como si éste fuera su propio cuerpo). En concreto nuestro Jmmp77 es descaradamente similar al Mad-3R Marauder de este juego.

Cuestiones relativas al diseño del modelo

Uno de los problemas más divertidos y complicados al mismo tiempo que implica el diseño de una de estas máquinas es que hay que crear un sistema de articulaciones que realmente funcione (una condición que no cumplen muchos dibujantes de Mechs).

Aunque la cabina y las armas principales de nuestro Mecha son muy similares a las del Marauder, vuestro seguro servidor intentó diseñar un cuerpo y un sistema de articulaciones diferente, entre otras cosas porque no acababa de visualizar correctamente el movimiento del Marauder.

El resultado no es, gráficamente hablando, nada espectacular, pero al menos es funcional. La cabina del piloto está sostenida por un largo cuello compuesto por varias piezas cuyos ejes de



giro nos permitirán enfocar dicha cabina hacia (casi) cualquier dirección. Aunque esto no es preciso ya que el Mech tiene varias cámaras- ojo situadas fuera de la cabina. Las armas principales son dos cañones gigantescos montados al final de dos brazos cuyas articulaciones hacen posible apuntar casi a cualquier parte, independientemente de la posición del resto del cuerpo.

Como arma secundaria hemos colocado un cañón doble en la parte superior de la torreta. Se supone que este cañón protegerá al Jmmp77 de ataques procedentes del cielo o de la retaguar-

día. (Sí, se echa en falta un lanzamisiles pero...) En cuanto a las patas, éstas pueden girarse lateralmente para cambiar de dirección con rapidez gracias a una pieza de giro situada en la cadera. Por último, cada pie está formado por cuatro pezuñas que permitirán al Jmmp77 sostenerse fácilmente en terrenos muy escabrosos.

Un ejemplo perfecto de los problemas que implica el diseño de articulaciones puede verse en esa inolvidable secuencia de Robocop en la que el tanque bípedo intenta bajar por unas escaleras.

Otra divertida cuestión relativa al diseño de estos artefactos es la “lógica” utilizada en su diseño. Dar algunos detalles sobre esto es algo casi obligado, así que avanzaremos que la fuente de energía de nuestro Jmmp77 procede de un microreactor de fusión cuya potencia se emplea para accionar los músculos artificiales de nuestro monstruo. La armadura del Mech está com-

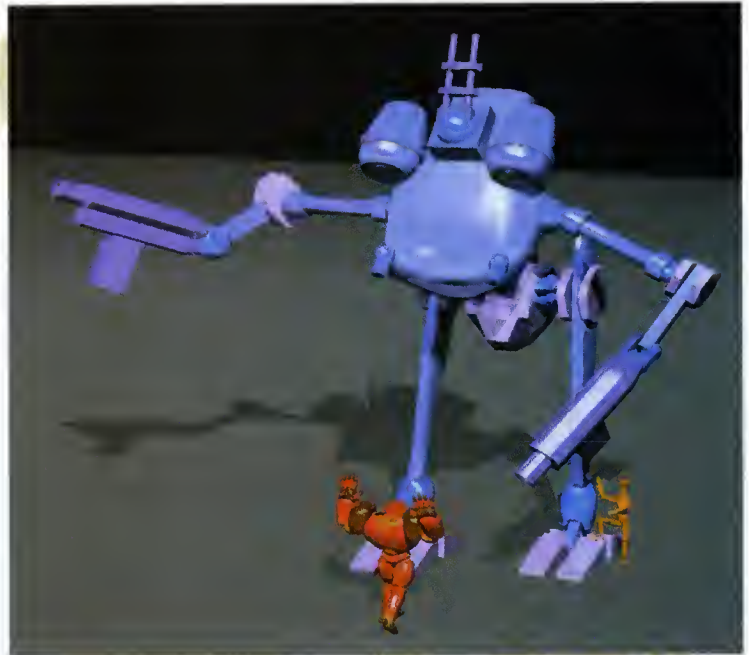
puesta por una aleación de impenetrabilite (¡Je!) que puede resistir a casi cualquier cosa y sus armas principales... (Nota del corrector de estilo: Suprimidas dos páginas de texto innecesario). ¿Críticas? El centro de gravedad es muy alto, hay un número excesivo de articulaciones, la estructura carece casi de blindaje -se pretendía que las articulaciones resultaran visibles-, etc, etc. (Nota de José Manuel Muñoz: Aquí el corrector de estilo insertó dos páginas propias con innecesarias críticas al modelo. Estas páginas han sido suprimidas por mí).



La edición de puntos

La edición de puntos es uno de los trucos más habituales a los que recurren los infografistas que emplean modeladores poligonales. Ni que decir tiene que la edición de puntos sólo será posible si el modelador empleado dispone de opciones que la soporten. Programas como 3D Studio o Imagine sí la permiten. ¿En qué consiste la edición de puntos? Bien, muchas veces los objetos generados con las herramientas de extrusión del programa o con otros métodos no tienen exactamente la forma que pretendíamos lograr. Esto puede deberse a limitaciones propias de las herramientas empleadas o a que, simplemente, la forma final del objeto no puede lograrse "del todo" con las tools usadas. Es en estos casos donde suele emplearse la edición de puntos que consiste en la modificación directa de la colocación espacial de los puntos del objeto. Para ello se emplean normalmente operaciones de traslación, rotación y escalación, aunque, adicionalmente, algunos modeladores permiten también otras operaciones.

Por supuesto, el estilo exacto de trabajo en la edición de puntos dependerá de la filosofía del modelador empleado, pero hay algunos aspectos comunes que comparten la mayoría de los modeladores. Quizá el más importante sea el detalle de que el trabajo se realiza sobre conjuntos de puntos. Esto es algo prácticamente obligatorio por dos razones: La primera porque el número de puntos a modificar dentro de un objeto puede ser muy elevado y la segunda porque -en muchas ocasiones- desearemos efectuar un desplazamiento simétrico sobre un grupo de puntos. Veamos seguidamente cómo funciona todo esto en Imagine.



Un caso práctico de edición

Prácticamente todos los objetos simples de que consta el Jmmp77 han sido tratados con edición de puntos. Puede que el caso más sencillo sea el del objeto que hace las veces de cabina. El lector puede comprobar las diferencias visibles entre cabina.frm, el objeto inicial creado desde el Editor de formas, y cabina.obj (el mismo objeto con los puntos correspondientes a los polígonos del morro desplazados y escalados). Para realizar las modificaciones entre ambas formas, leed el objeto cabina.frm desde el Editor de Detalles y seleccionadlo (desde el modo de objetos). Después activad el modo de tratamiento de puntos (opción "Pick Points" del submenú "Mode") y marcad el modo de selección "Drag Box" dentro de "Pick Method", en el submenú "Mode". Al cambiar a "Pick Points" los puntos del objeto seleccionado se

remarcarán en amarillo. Al seleccionar los puntos deseados del morro de la cabina englobándolos dentro de la caja de selección, estos tomarán un color rosado. (Nota: Pueden sumarse más grupos de puntos a una selección empleando varias cajas). Cuando esto suceda, y si no os habéis equivocado al seleccionar los puntos a editar (en cuyo caso tendréis que pinchar con el ratón y efectuar una nueva selección), se podrán efectuar las operaciones de edición.

En el caso que nos ocupa, lo primero será trasladar los puntos del morro para sacarlos un poco del objeto. Para ello pinchad primero en el botón de movimiento (de la fila inferior de botones) y luego, como interesa restringir el movimiento de los puntos al eje Y, desactivad el movimiento de los otros ejes pinchando sobre sus botones. Recordad que en Imagine, por defecto, las transformaciones de movimiento y escala-

ción se efectúan sobre todos los ejes. Al trabajar con puntos u objetos suele ser interesante restringir el alcance de las operaciones a uno o dos ejes. Al ordenar una transformación espacial de movimiento o escalación, todos los ejes estarán activados. Para restringir la operación a uno o más ejes habrá que pinchar sobre los botones X, Y o Z, según sea el caso.

Hecho esto, pinchad y arrastrad con el ratón en la ventana Right para des-

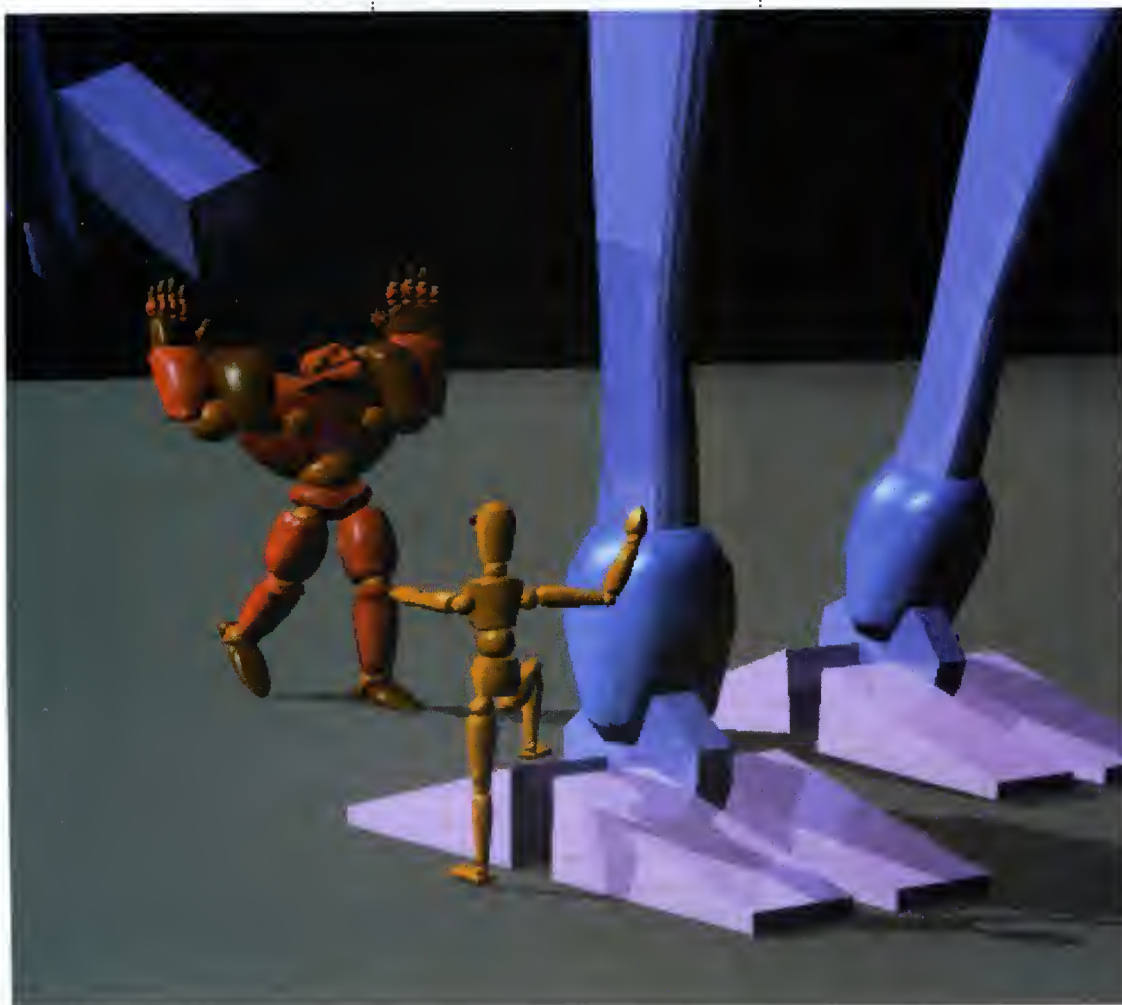
plazar un poco los puntos fuera del morro del objeto. Después, manteniendo la misma selección de puntos, pinchad sobre el botón de Escalación y arrastrad con el ratón de modo que los puntos se expandan. De esta manera “abriremos” un poco el morro de la cabina.

Luego podéis seleccionar otro grupo de puntos más reducido en el exterior de la cabina y desplazarlo hacia adentro, con lo que se creará una depresión dentro de ésta.

Nota: En caso de meteduras de pata siempre podéis hacer ALT-U -undo- o bien borrar el objeto y volver a importar a cabina.frm.

Creando un objeto por edición de puntos

La modificación de objetos contruidos con otras herramientas no es la única utilidad de la edición de puntos. El trabajo directo desplazando y escalando puntos puede resultar un tanto te-

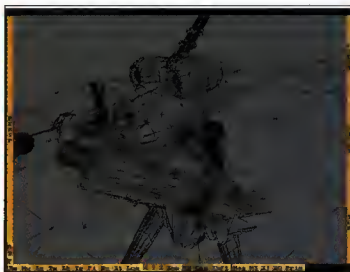




dioso al principio pero se trata de un sistema tan directo y preciso de modificar la forma de los objetos que uno acaba prefiriéndolo a otros sistemas teóricamente más cómodos, incluso en el caso de la creación de objetos nuevos. Veamos un ejemplo.

Uno de los objetos simples que componen nuestro Jmmp77 es una pieza parecida a la cabeza de un tornillo que se encuentra -sola o en pareja- en varias de las articulaciones del mech. Esta pieza se creó generando primero un disco simple (opción "Disk" de la ventana de primitivas del botón "Prim"). Una vez creado el disco se procedió, después de seleccionar el objeto, a extrusionarlo (opción "Extrude" de la ventana "Mold", cuyo acceso se halla en el submenú "Functions") para conseguir la forma general inicial de la pieza (un cilindro).

Después, desde la ventana frontal y entrando en el modo "Pick Points", se seleccionaron los puntos de cada sección para proceder a su escalado y desplazamiento. Haciendo esto, y cuando ya casi se había concluido con el diseño de la pieza, resultó que hacía falta una sección más para crear la depresión interna de la pieza. Para lograr dicha nueva sección se entró en el modo "Pick Faces" (selección de caras) y se seleccionaron los puntos del centro. (Nota: Seleccionar caras con la caja de "Drag Box" puede resultar bastante fastidioso ya que es muy fácil seleccionar más caras de las necesarias. Para evitar este problema podemos intentar la selección desde otra ventana o bien usar el lazo de selección -opción "lasso" de "Pick Method"-). Las aristas de las caras seleccionadas tomaron un color azul. Entonces se se pinchó sobre la



opción "Fracture", en el submenú "Object". Fracture crea nuevas caras subdividiendo el espacio ocupado por los polígonos seleccionados. De esta manera apareció un nuevo conjunto de puntos que, después de volver al modo "Pick Points", fueron debidamente escalados y desplazados.

La opción fracture suele emplearse mucho para suavizar contornos de objetos que se han creado con un número excesivamente reducido de polígonos. (Podemos emplear fracture para subdividir caras y líneas). También podemos emplear fracture, como en este ejemplo, para crear puntos nuevos en la superficie de un objeto.

Problemas y trucos de la edición de puntos

La edición de puntos también presenta algunos problemas. El más importante es, desde luego, realizar correctamente la selección de los puntos. A veces, como en el ejemplo precedente, esto es muy sencillo pero en otras ocasiones, y debido a la forma del objeto seleccionado, se trata de una operación bastante difícil, o al menos lo sería si imagine no contara con la opción "Hide Points". ¿Y por qué puede haber problemas en relación a esto?

Básicamente, debido a la forma en que se seleccionan los puntos. Normalmente usaremos una caja de arras-

tre con el ratón manteniendo pulsada la tecla shift (modo múltiple). La caja incluirá en la selección no sólo a todos los puntos visibles dentro de ella pertenecientes a la sección que nos interese, sino también a todos los puntos del objeto que tengan dos de sus coordenadas 2d comprendidas dentro de las coordenadas de la caja. O sea que probablemente se incluirán también puntos que no nos interesen o que ni siquiera serán visibles dentro de la ventana de trabajo en activo debido a que tienen las mismas coordenadas 2d de otros puntos más cercanos al espectador. En otras palabras: La ventana de selección tiene dos dimensiones visibles en la ventana de trabajo desde la que es dibujada, pero realmente se extiende perpendicularmente "entrando" en la ventana de trabajo para seleccionar los puntos.

Si el lector se siente un poco confuso con la explicación precedente imagine el siguiente caso: se crea un plano de 10*10 puntos con la opción "Plane" de la ventana de primitivas (botón "Prim") y después se extrusiona el plano obtenido usando la opción "extrude" de la ventana "mold" y ordenando 10 secciones. Nuestro nuevo objeto será un cubo -si hemos especificado una longitud para la extrusión igual al largo de uno de los lados-, y tendrá 10*10*10 puntos. Si ahora seleccionamos el objeto y, pasando al modo "Pick points", nos situamos en la ventana Front y creamos una caja de selección de puntos que abarque a 2*2 puntos, los puntos seleccionados no serán 4, sino 40 (2*2*10). (Lo sucedido nos resultará evidente si examinamos los puntos rosados en las tres vistas principales de trabajo). Supongamos

que lo que se pretende es precisamente seleccionar a los 4 puntos más cercanos visibles dentro de la caja en la ventana Front, o sea los más cercanos al espectador. ¿Cómo hacerlo?

La única forma es utilizar la opción "Hide Points" del submenú "Mode". Para ello, desde el modo de puntos, pulsaremos esta opción y crearemos una caja de "ocultación" que englobará a los puntos cuyo dibujo y uso queremos desactivar temporalmente. Es importante tomar nota de que, en este caso, no hay que apretar la tecla de shift (modo múltiple). ¿Dónde colocaremos la ventana de ocultación?

En nuestro ejemplo, y puesto que queremos seleccionar un grupo de puntos que están en la cara del cubo más cercana al espectador, en la ventana Front, lo adecuado será crear dicha caja en la ventana Top o en la Right. La ventana debiera abarcar a todas las secciones del cubo menos a la necesaria para hacer la selección. Una vez ocultos los puntos, volveremos a pinchar en "Pick points" para hacer la selección de los 2*2 puntos deseados. Naturalmente los puntos desaparecidos siguen estando ahí y volverán a aparecer si cambiamos al modo de objetos ("Pick objects"). El uso de la opción "hide points" es prácticamente impres-

cindible en objetos con mallas de puntos muy densas.

Por último hay que tener en cuenta otro posible problema: En muchos objetos creados desde el Editor de Formas la disposición de los puntos no es simétrica en los distintos ejes –aunque la forma general del objeto sí lo sea–. Esto se debe a que los puntos equivalentes de las distintas formas 2d no tienen

alas de un avión). Este problema puede ser parcialmente enmendado usando fracture para crear nuevas caras con las que intentaremos nivelar un poco las diferencias; pero esto tampoco es demasiado cómodo.

Problemas con el redibujado

Cuando se está trabajando con puntos o cuando se está estableciendo una

jerarquía de objetos o en medio de otros procesos, podemos tener ciertos problemas con el redibujado de objetos.

La verdad es que el autor de este artículo desconoce, en el momento de escribir estas líneas, si dichos problemas se deben a pequeños bugs de la versión 3.0 de Imagine o a valores incorrectos que pueden cambiarse en alguna parte; por ahora me inclino a creer

que se trata del primer caso. De cualquier manera estos fallos en el redibujado pueden ser bastante molestos.

Puede suceder por ejemplo que al retornar desde un modo de selección a otro –y sobre todo si usamos "hide points"– los objetos no se redibujan en el modo adecuado, que quede basura o que ciertos objetos permanezcan invisibles. Esto podemos arreglarlo usando la opción "Redraw" (redibujado) del submenú "display" o maximizando



muchas veces la misma altura. (A veces esto sencillamente es imposible si queremos lograr una forma dada). Cuando esto suceda tendremos problemas en el Editor de Detalles si pretendemos realizar cambios simétricos en la forma del objeto editando puntos de la manera ya descrita. (Nota: Por cambios simétricos nos referimos a alteraciones idénticas en zonas equivalentes de objetos con, por ejemplo, simetría bilateral, como puede ser el caso de las



una ventana o cambiando entre uno y otro modo de selección.

Articulaciones y problemas con los ejes

Un punto sobre el que no hicimos demasiado hincapié en el capítulo 4 del número 5 de Rendermanía fue el de la importancia de los ejes en la construcción de las jerarquías de movimiento del modelo. Como recordaréis, en Imagine al crear un objeto cualquiera siguiendo un procedimiento cualquiera se crean siempre unos ejes que se representan con tres líneas que cruzan un punto y sobre las que pueden verse las letras X, Y y Z. Estos son los ejes locales de cada objeto y son los que se emplean en las jerarquías de objetos para colocar las posturas de las extremidades de los modelos empleando rotaciones.

Por supuesto, esto quiere decir que la preparación de las posturas del modelo sólo será posible si los ejes de cada objeto se hallan en los puntos adecuados. Considérese el caso del cuello del modelo de un ser humano virtual.

Al construir las jerarquías haremos que la cabeza sea un objeto hijo del objeto cuello. Pues bien, si el eje del cuello está bien colocado (en el centro del mismo) podremos —pinchando en “Pick Groups” y seleccionando el cuello— hacer girar a la cabeza correctamente. Sin embargo si el eje está mal puesto puede ocurrir que, al efectuar la rotación, ¡la cabeza se salga del cuello!

La posición final de los ejes de un objeto recién creado no tiene por qué coincidir con la posición final que habrán de tener dichos ejes cuando el objeto ocupe su lugar dentro del modelo. Puede ocurrir que los ejes estén muy



descentrados e incluso que queden fuera del objeto. Esto puede ocurrir fácilmente en el Editor de Formas, en donde al desplazar los puntos de las formas 2d puede suceder que no tengamos en cuenta la posición de los ejes dentro del objeto. También puede ocurrir esto si creamos un objeto en el Editor de detalles siguiendo el procedimiento de dibujar primero los puntos y luego las caras para, finalmente, extrusionar la forma 2d. Resumiendo: Para que la manipulación de los modelos articulados funcione tendremos que situar adecuadamente los ejes después de haber colocado espacialmente a cada objeto simple en su correcta posición dentro del modelo. Esta manipulación de los

ejes de un objeto sin alterar la posición espacial de este podemos llevarla a cabo seleccionando el objeto deseado y pulsando shift-M (desplazamiento). Haciendo esto podremos trasladar los ejes directamente en alguna de las ventanas de trabajo.

También, si deseamos una mayor precisión, podremos invocar con este fin a la ventana de transformaciones y llevar a término la operación marcando el flag “Axes only” (sólo los ejes).

Por último debemos señalar que es conveniente imaginar (al menos en casos como éste) cómo va a moverse el modelo antes de diseñar los objetos. Si no tenemos cierta previsión pueden acabar sucediendo cosas tales como que las distintas piezas se superpongan en medio de algún giro o que algunos miembros sean excesivamente cortos o largos como para cumplir su cometido.

El acabado

Finalmente conviene recordar un detalle importante de cara a ahorrar trabajo inútil: Es conveniente dar atributos a los objetos en cuanto la forma de estos se da como acabada. Si no obramos de esta manera tendremos probablemente que repetir el trabajo varias veces. (Por ejemplo para indicar superficies flat en objetos mirror o en derivados).

El Mech

Cread un proyecto y luego un subproyecto con los nombres que queráis. Luego volcad el contenido del directorio objects en el subdirectorio del mismo nombre dentro de vuestro nuevo proyecto. Después, seguid los pasos normales desde el Editor de escenas si queréis preparar el primer render.

NOTA:

Acerca de los objetos “MIRROR”

En el número 5 se explicó un truco para crear copias espejo de objetos. Recordemos que una copia espejo es un objeto con simetría opuesta a la del original. Por ejemplo, las alas de un avión. Pues bien este truco no funcionará a menos que marquemos el flag “World” en la ventana de Transformaciones. Pedimos perdón por esta imperdonable omisión.



Lo mejor de dos mundos



Una prueba de portada con niebla sin transmisión de luz.

Para la portada del presente número era obligatorio emplear el modelo de ejemplo de este mes. Luego se decidió, además, realizar el render final desde Pov. Decidido esto, pasaron a considerarse diversos métodos para realizar la conversión Imagine-Pov. Fue en este momento cuando el autor se acordó de

3ds2pov, la conocida “tool” creada por Steve Anger. Examinando algunos de los ficheros “convertidos” por 3ds2pov, el autor de estas líneas concibió un método para resolver los típicos problemas de traducción. Un método más preciso y sencillo que el explicado en el número 1 de Rendermanía. Sin embargo (¡ay!), 3ds2pov únicamente convierte escenas de 3D Studio (al formato de Pov). Hasta el momento no hemos

En el número anterior integramos en la portada elementos de Pov e Imagine. Hoy volveremos a hacer esto para explicar el “traslado” de un modelo de Imagine a Pov. Por supuesto, existen muchos caminos para realizar una conversión de este tipo pero el método actual es, cuanto menos original, ya que implica la mediación de 3D Studio.

encontrado ninguna utilidad que haga lo propio entre Imagine y Pov.

Explicaremos el siguiente método porque 3ds2pov es de libre distribución y porque 3D Studio es un programa conocido por un gran número de lectores. (Además cabe imaginar que entre ellos habrá quien prefiera utilizar Imagine o que quiera utilizar ambos paquetes). Este método es algo indirecto pero muy sencillo.



La grabación DXF

Comenzaremos la sesión en el Editor de Detalles de Imagine cargando el modelo que deseamos trasladar al universo de POV. Este modelo habrá de ser guardado en formato DXF para que pueda ser importado por 3D Studio. Una vez que esto sea hecho, grabaremos el modelo en formato 3ds y pasaremos a emplear 3ds2pov para realizar la conversión final. A grandes rasgos éste es el método, pero hay que tener en cuenta una serie de detalles si queremos que la conversión sea lo más fiel y cómoda posible.



Comprobando la exactitud de la escala.

Lo primero que tenemos que tener en cuenta es que normalmente un modelo está compuesto de muchas piezas con diferentes atributos de superficie. En el caso de nuestro Jmmp77 queremos conservar, al menos, el color de cada pieza. ¿Cómo hacer esto?

Para conseguirlo nos veremos obligados a grabar varios ficheros Dxf, no uno sólo. Afortunadamente no será preciso grabar un .dxf por cada objeto simple. En lugar de esto grabaremos un archivo dxf con cada grupo de objetos que presenten los mismo atributos de

superficie. Para ello efectuaremos un quickrender para tomar nota de las piezas que tienen el mismo color. Hecho esto, y desde el modo de selección de objetos, iremos pinchando en ellos manteniendo pulsada la tecla shift (modo múltiple). Cuando todos las piezas deseadas estén seleccionadas, las agruparemos pulsando sobre "Group" en el submenú "States". Previamente hay que desagrupar el modelo entero con "Ungroup". Hecho esto, entraremos en el modo de grupos, seleccionaremos el grupo recién creado y lo grabaremos como dxf ("Save Dxf", en el submenú "objects").

Luego borrarémos el grupo con "Delete" y, después de efectuar otro "Quickrender" si es preciso, repetiremos todo el proceso con otro grupo de objetos de idéntico color. Continuaremos con este ciclo hasta

que no quede en pantalla ninguna de las piezas del modelo.

Importando el DXF

El siguiente paso será ir leyendo los ficheros .dxf desde 3D Studio. Al primer grupo lo cargaremos con Load pero para los siguientes utilizaremos Merge ya que de no hacerlo así borraríamos los objetos ya leídos. Cuando todos los ficheros .dxf se hayan cargado tendremos el modelo completo en 3D Studio. (En las pruebas se usó la opción "layer", al importar Dxf).

Seguidamente hay que preparar el modelo antes de guardarlo como fichero 3ds. Si grabamos ahora, al usar 3ds2pov nos percataremos, al examinar el fichero .pov resultante, de que sólo se ha definido una única textura para todos los objetos. Para evitar esto tendremos que proceder de la siguiente forma: Antes de cargar cada fichero .dxf desde 3D Studio, pulsaremos sobre la opción "choose" (elegir), en el menú "Surfaces" y elegiremos una textura cualquiera tomando nota del nombre de la misma. Después cargaremos el grupo dxf que corresponda y le asignaremos la textura leída (con "Assign"). Luego volveremos a cambiar la textura en curso eligiendo otra nueva con "choose" y cargaremos el siguiente archivo .dxf. Procederemos así hasta que cada grupo de objetos tenga su propia textura. Y lógicamente deberemos tomar nota de a qué grupo de objetos hemos aplicado cada textura.

La orientación espacial del modelo

Como recordarán quienes leyeron el número 1 de Rendermanía, el sistema de ejes de otros programas no tiene por qué coincidir con el de POV y ese es el caso de Imagine y de 3D Studio. Para que el modelo sea almacenado en el fichero 3ds con la orientación con la que trabajamos normalmente en POV (o sea: de pie sobre el plano X-Z y con la cabina orientada paralelamente al eje Z y la cara mirando al lado negativo de éste), tendremos que efectuar sobre el algunas operaciones de rotación. Deberemos hacer que el modelo esté de pie y mirando hacia nosotros en la ventana Top de 3D Studio. Además el modelo deberá quedar de perfil en la ventana

Left, con el eje cabeza-pies formando una línea paralela a la de esta ventana y con la cabeza apuntando al lado izquierdo de dicha ventana y con el morro hacia abajo. Una vez conseguido todo esto grabaremos el fichero 3ds.

Aparte de esto necesitaremos dos datos: información acerca de la altura a la que quedan los pies del modelo

y la altura total del mismo. Para lo primero crearemos un plano de dos dimensiones (o importaremos el plano del suelo de Imagine) que rotaremos y desplazaremos para colocarlo como suelo a los pies del mech, una vez que hayamos dado a éste la orientación antes descrita. (El plano será perpendicular al Mech puesto de pie y equivale al suelo normal de Pov). Luego crearemos una caja alargada cuya base se apoyará en el suelo creado y cuyo techo llegará hasta el techo de la cabina del mech. Grabaremos la caja y el suelo como un fichero 3ds aparte (llamado orienta.3ds).

Trabajando sobre el fichero POV

Ahora procesaremos los dos ficheros 3ds con 3ds2pov. Por si alguien no lo recuerda, bastará con teclear

```
3ds2pov nombre_fichero.3ds nombre_fichero.pov
```

3ds2pov creará dos ficheros a partir del .3ds con el mech; un archivo .inc donde se almacenarán las sentencias



Otra posible semilla para la ciudad.

triangle y smooth_triangle con la malla poligonal de los objetos y un fichero .pov donde irá la correspondiente orden incluye (del fichero .inc) y donde se guardará la “conversión” de las texturas de 3D Studio al sistema Pov. Como recordaréis, 3ds2pov realiza una “traducción” para, en la medida de lo posible, convertir los atributos de superficie al lenguaje escénico de Pov.

En este caso, esta conversión no nos interesará ya que lo que pretendemos es escribir manualmente los atributos de superficie que tenía el modelo en Imagine. Para ello consultaremos los valores de color que tenía cada grupo de piezas. Estos colores se describen con una tripleta rgb cuyos valores oscilan entre 0 y 255. En Pov, por el contrario, la intensidad de cada tono se describe con un valor flotante que oscila entre 0 y 1. Por tanto para “traducir” cada tono tendremos que dividir cada valor entre 255. Así, por ejemplo, los valores rgb de Imagine <132, 103, 186> pasarán a ser <.517, .404, .729> en Pov. Lógicamente, después de haber tomado nota de los colores originales de Imagine, tendremos que escribir su

“traducción” en las definiciones de color de las texturas del fichero .pov.

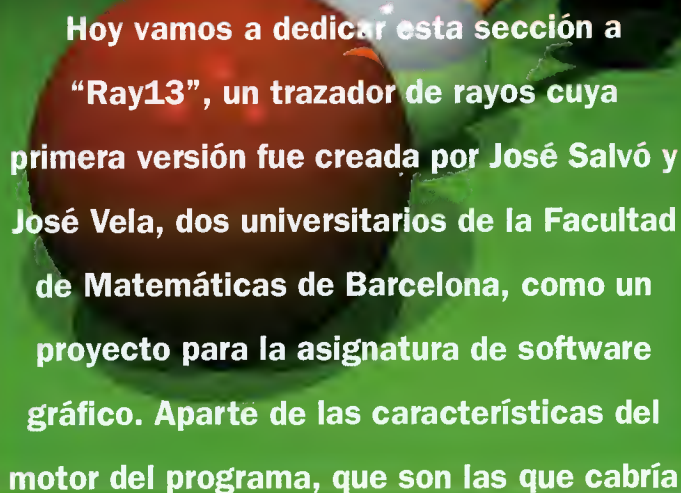
Por último tendremos que escalar y situar el mech. Echemos un vistazo al fichero pov resultante de la conversión de orienta.3ds. Veremos que 3ds2pov ha incluido unas líneas informativas con datos acerca del punto 3d donde se halla el “centro de gravedad” de cada objeto (Center X, Y y Z) y las longitudes en cada eje (Length X, Y y Z). Hay un objeto que tiene una longitud de cero en el eje Y, es el plano-suelo sobre el que están los pies del mech, por tanto el valor de longitud de ese suelo en el eje Y indica la altura a la que están los pies del mech (fácil ¿no?). Ahora vayamos con la escala. Queremos que el Mech tenga 15 metros de alto y vamos a emplear los pov-edificios del número anterior en los que 1 unidad equivalía a un metro. De esto se deduce que para obtener la escala correcta tendremos que dividir 15 entre la longitud Y del mech y aplicar el valor resultante en la escala. Resumiendo:

- 1) Declaramos una unión con todos los objetos del Mech
- 2) Escribiremos una operación translate restando la altura del mech (con el valor hallado con el suelo).
- 3) Pondremos una orden de escalado al mech. Si la longitud Y de éste es, por ejemplo, de 516 la orden será scale<.029, .029, .029>.

Luego incluiremos los edificios. Como veréis, con este método nos ahorraremos establecer la escala de los modelos para la escena de Pov desde 3D Studio, lo cual puede ser bastante molesto e impreciso. Sobre todo si queremos exportar varios modelos.



En el CD "Ray13"



Hoy vamos a dedicar esta sección a "Ray13", un trazador de rayos cuya primera versión fue creada por José Salvó y José Vela, dos universitarios de la Facultad de Matemáticas de Barcelona, como un proyecto para la asignatura de software gráfico. Aparte de las características del motor del programa, que son las que cabría

esperar de un trazador de rayos, "Ray13" presenta dos aspectos muy interesantes: el documental –los autores explican los pormenores del desarrollo y las fórmulas empleadas– y el hecho de que "Ray13" es un producto que aún se halla en fase de desarrollo.

Por "fase de desarrollo" nos referimos al hecho de que José Vela confía en que un buen número de rendermaniacos vea el programa y envíe sugerencias y opiniones y, en suma, declare si le parece que "Ray13" es un producto lo suficientemente interesante como para justificar la aparición de nuevas versiones. Por esta razón, y para dar ejemplo, vuestro seguro servidor enfocará la mayor parte del presente artículo como una carta abierta dirigida a los desarrolladores de "Ray13" (o a cualquier programador con capacidad, tiempo y ganas para hacer un programa de este tipo). Antes, sin embargo, veamos qué incluye el producto que podéis encontrar en el CD.

Instalación y funcionamiento

Al descomprimir el fichero arj (con la opción x) aparecerá una serie de subdirectorios con imágenes de ejemplo (demo), un fichero en el formato de Microsoft Word 6.0 (doc), ficheros escénicos de ejemplo (files.r13) y los fuentes del programa en Borland C 3.1 (fuentes). El propio trazador y varias utilidades adicionales aparecerán en el directorio raíz donde se efectúe la descompresión.

"Ray13" es, como dice José Vela, un trazador de rayos en la línea de POV (los fuentes de POV son citados en la bibliografía del doctoral). Funciona mediante órdenes impartidas desde



DOS y procesa ficheros de generación de escenas escritos en un lenguaje propio bastante rudimentario. Dentro de dichos ficheros se incluyen instrucciones no sólo para el diseño de las escenas sino también para controlar el proceso del render. Así podremos indicar a "Ray13" si queremos o no sombras, reflexiones, refracciones, estadísticas, interrupciones por teclado, etc. En cuanto a las primitivas del lenguaje, "Ray13" soporta planos, esferas, elipsoides y cuadrículas sobre las que podremos efectuar las conocidas transformaciones espaciales. Ordenar un render es tan sencillo como escribir:

```
ray13 fichero_entrada.r13 fichero_salida.jvr
```

El fichero .jvr resultante podremos verlo con el visor "Viewjvr" o transformarlo a tga con la utilidad "Jvr2tga" para verlo después con un visor más convencional. En la primera prueba, el programa devolvió un mensaje indicando que el modo gráfico pedido no estaba disponible. Este problema desapareció al cambiar los parámetros de la sentencia "Modo Gráfico" que estaban a "640,480,24" por "640,480,16". Sin embargo, al realizar otra prueba con un valor de 8 bits por pixel (256 colores), el resultado fue una imagen llena de colorines (¿un fallo en el algoritmo de reducción de color?)

Carta de un usuario al autor de "Ray13" Versión 3.0

La opinión del autor del presente artículo es que "Ray13" podría llegar a ser un trazador de rayos de primera fila. Obviamente, le faltan aún muchas cosas, pero parece indiscutible que tienes los conocimientos para crear un producto de gran interés. Que dispongas

del tiempo suficiente para ello ya será otra cuestión ya que -aunque estoy muy lejos de tener tu nivel matemático- me huelo que sólo la implementación de un buen lenguaje escénico ya se te llevaría bastante más de los dos meses (!) que dices que os llevó la primera versión.

La versión 3.0 de "Ray13" es, por ahora, poco más que un buen motor de render. Para que llegue a convertirse en un programa de utilización masiva como "POV" o "Polyray" necesitará ante todo que los usuarios disfruten gene-



rando escenas con él. Pero, ¿cómo crear estas escenas? Si sigues periódicamente el Foro del Lector te habrás dado cuenta de que el nivel gráfico de los usuarios ha subido mucho con respecto al de hace unos años. Los usuarios emplean las más diversas herramientas como modeladores, lenguajes complejos, conversores, etc. para crear sus escenas. Esto quiere decir que el futuro de "Ray13" pasa por que ofrezcas alguna vía para que los usuarios pue-

dan generar escenas de gran complejidad con él.

Los tiempos en los que bastaba con generar escenas a base de esferas ya han quedado atrás. Estamos de acuerdo en

que una escena de ese tipo puede servir a la perfección para ilustrar las capacidades del motor, pero el usuario necesita -y exige- ciertas herramientas de soft mínimas para desarrollar sus escenas. Por lógica, las posibles versiones futuras de "Ray13" deberían pasar por uno de estos puntos: 1) Mejorar el lenguaje escénico hasta darle una complejidad similar o superior al de "POV". 2) Implementar un modelador. 3) Crear una utilidad que permita que "Ray13" pueda interpretar escenas de otros programas.

Las dos primeras líneas de acción requerirían (imagino) una gran cantidad de tiempo y esfuerzos sin tener la seguridad de conseguir un resultado realmente impactante. La tercera, por otro lado, haría que "Ray13" dependiera excesivamente de otros programas. Probablemente cada usuario te enviará su propia lista de sugerencias de mejoras y consejos para dar una "personalidad" propia a tu programa; a continuación te comento las mías.

Sugerencias de un usuario

Sería estupendo disponer de un trazador de rayos que aunara las ventajas propias de los modeladores como "Imagine" o "3D Studio" con las capacidades de un lenguaje escénico similar al de "POV". Que yo sepa aún no se ha hecho nada como esto. Mis sugerencias son:

1) Olvídate de la creación de objetos para el lenguaje escénico de "Ray13". En lugar de eso, implementa lo mínimo para que "Ray13" pueda digerir



modelos poligonales creados con otros programas (modelos, no escenas). Estos modelos deberían ser tratados y procesados después por "Ray13". Sería interesante que el lenguaje permitiera manipular jerárquicamente los objetos constituyentes de los modelos. No sólo para crear posturas, sino para permitir variaciones al azar entre posturas definidas de un mismo modelo. Además, lo deseable sería que la descripción de un modelo sólo se almacenara una vez y que las "variaciones" del mismo sólo se guardaran como matrices con la descripción de los cambios con respecto al modelo original. Esto permitiría crear escenas con miles de objetos complejos no exactamente idénticos sin necesidad de sufrir por la falta de memoria (lo cual es uno de los problemas de "POV"). Naturalmente, el lenguaje debería admitir bucles, ifs y funciones rand para tratar la colocación de estos modelos y permitir la creación de escenas tipo hormiguero (que son difíciles de hacer con un modelador).

2) Abundando en lo anterior sería interesante que el usuario pudiera crear arrays de objetos para comprobar la distribución espacial de los modelos. Y también sería interesante que el volumen interior de los mismos fuese reconocido como tal por el programa a fin de resolver cuestiones como la colocación de objetos en superficies no lisas y los choques entre estos, con vistas a realizar animaciones (ya sé que esto último es tremendamente difícil). Resumiendo: sería muy interesante que el lenguaje se enfocara más a la disposición y a la animación espacial de los modelos (importados) que a su descripción.

3) Incorpora todos los efectos atmosféricos que puedas. Sé de muchos



lectores que siguen empleando "Povafx" —una versión retocada de "POV 2.0" creada por Marcos Fajardo— en vez de "POV 3.0". Y lo hacen sólo para seguir empleando los magníficos efectos atmosféricos de Marcos. Este es un punto al que los usuarios dan mucha importancia.

4) Implementa un sistema automático de bounding-slab. Una de las ventajas de los lenguajes escénicos (con sentencias de programación) es que permiten definir muchos elementos con pocas líneas de código. Si esta ventaja se pierde por un tiempo de render excesivo (me imagino que la velocidad del render de "Ray13" caerá como una piedra si defines una escena con bastantes elementos; corrígeme si me equivoco). Como pov-usuario me gustaría que el bounding automático de "POV" fuera más eficiente con los objetos csg (en el número anterior pretendía crear una ciudad completa con edificios que tenían roturas csg colocadas aleatoriamente en cada estructura; pero al final, y por



culpa de los tiempos de generación, tuve que eliminar las operaciones csg).

5) Haz que el lenguaje permita crear texturas algorítmicas. ¡Las de "POV" son maravillosas!

6) Si "Ray13" va a seguir siendo un producto de DOS, quizá debieras emplear un compilador como Gnu o Watcom por las cuestiones de la velocidad del ejecutable y el direccionamiento lineal de memoria.

7) Haz que "Ray13" genere ficheros tga directamente.

Conclusiones

Desde luego implementar estas sugerencias no es precisamente pan comido (sobre todo las correspondientes a los puntos 1 y 2). Pero aumentar la potencia del lenguaje escénico hasta conseguir algo parecido al lenguaje de "POV" tampoco lo es (y por este último camino "Ray13" acabaría pareciéndose demasiado a "POV"). Además, crear un modelador potente tampoco es un camino fácil. En resumen: "Ray13" es, en su versión 3.0, un excelente punto de partida, pero ahora, si quieres lograr que se acabe convirtiendo en un producto interesante para el usuario, debes dotarle de una "personalidad" propia en el apartado de construcción de las escenas (por uno u otro camino). Las mejoras en el aspecto de la potencia del motor serán siempre bienvenidas, pero si no se complementan con un sistema que permita crear escenas complejas con más facilidad...

En suma, tu trabajo es impresionante (¡no lo dejes aquí!). Confío en que no te tomes a mal mis sugerencias y el que no haya hecho una crítica más "científica" de "Ray13", pero es que las fórmulas me intimidan un poco...



Nota importante. Podéis remitirnos vuestros trabajos o consultas, bien por carta a la dirección que figura en la segunda página de Pcmánia, o vía e-mail a rendermania.pcmania@hobbypress.es

Batallas virtuales

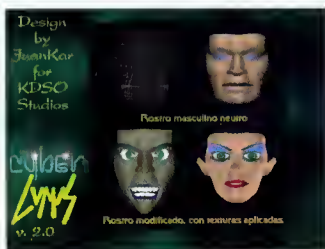
Aunque el asunto de la batalla virtual entre los humanos y los orcos empezó como una broma, el continuo envío de modelos para reforzar las huestes de uno u otro bando por vuestra parte nos obliga a declarar que la prometida batalla se celebrará muy pronto. Mientras tanto, hoy podremos, entre otras cosas, echar un vistazo a algunos de los modelos que participarán en la tan esperada contienda: un par de nuevas Catapultas y hasta un mago.



Juan Carlos Jiménez Méndez de Kdso Studios es recordado aquí por su trabajo Cyber Lumis, un modelo femenino creado con 3D Studio del que este autor prometía una futura versión hecha con blobs. ¡Pues bien, aquí está! La nueva versión ha sido creada con Blob Sculptor 2.0 y aún no es perfecta, según afirma el propio Juan Carlos, pero espera remediar las imperfecciones con la próxima versión de Blob Sculptor. ¿Que dónde se encuentra dicha nueva versión y cómo conseguirla? Bueno, hoy no lo sabemos pero quizá podamos responderte en el próximo número.

También es de agradecer el impresionante mago que ha enviado Juan Carlos para reforzar a los orcos. El tronco principal del mismo y otras partes se han logrado modificando vértice a vértice un samurai obtenido en el infograf 96. El resto de las partes se han creado a base de extrusiones. En cuanto al rostro, como en Cyber Lumis, se obtuvo modificando un rostro masculino neutro al que se añadieron texturas. (Si alguien cree que esta técnica carece de mérito, que eche un vistazo a la imagen con las caras originales y las finales).

Juan Carlos afirma que su Nigromante no está demasiado depurado pero el resultado, opinamos aquí, es fabuloso. ¿¡Qué has cedido a Cyber Lumis para que sustituya al Cyberdemon del Xmanga de Marg!?! A ver cuándo lo acabáis... Felicitaciones por tu magnífico trabajo y anótate un tirón de orejas con retorcimiento triple por no enviar tu carta al foro.





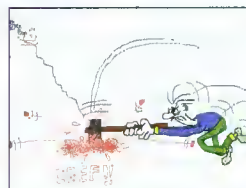
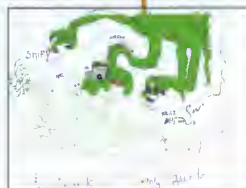
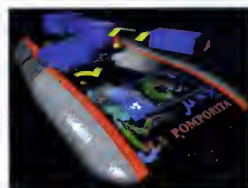
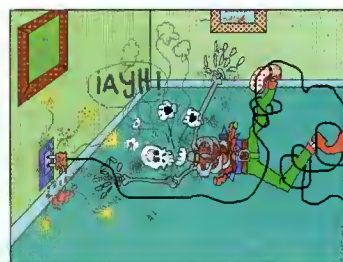
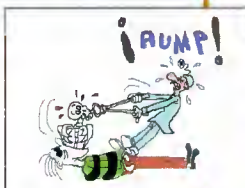
El Foro del Lector

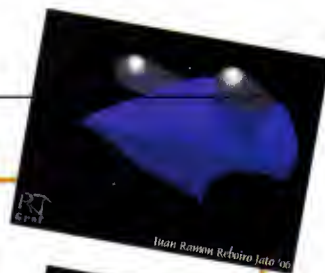


Nasser Afkir Torres regresa de nuevo a estas páginas con un estupendo trailer que servirá para transportar la grúa que ya envió anteriormente. Ambos objetos están hechos con 3D Studio y podéis encontrar el proyecto y la carta correspondientes en el foro.

Juan Pacheco Guerrero está trabajando en una animación –de la que ya tiene unos 10 minutos de rodaje– basada en la película Tron. Juan se queja de la falta de espacio en los Cds

para publicarla. Se comprende que unos 60 megas te parezcan pocos para una película de 20 minutos pero en el CD han de entrar muchas cosas y resulta difícil saber a priori de cuanto espacio libre va a disponerse. Por cierto anótate un tirón de orejas, ya que en tu carta (en el foro) has hablado de casi todo menos de los detalles de tu trabajo.



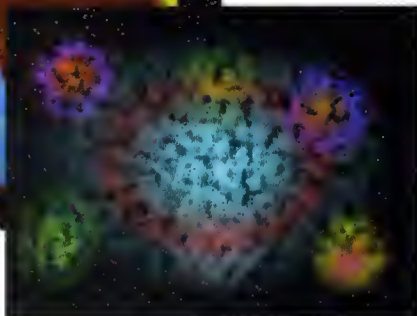


Los hermanos **Reboiro Jato, Roberto y Juan Ramon**, nos envían diversos trabajos realizados con Pov y Moray. Las vicisitudes del trabajo de ambos artistas pueden ser seguidas en su carta publicada en el foro. En ella Roberto pregunta a vuestro seguro servidor si "sabe algo sobre los derechos que tenemos los autores sobre nuestras escenas". Pues bien, el mayor problema consiste en demostrar que tu trabajo es efectivamente tuyo. Si puedes hacerlo, podrás denunciar cualquier uso que alguien haga de él sin tu permiso. Para ello puedes informarte en el registro de la propiedad intelectual. Aunque, si no quieres molestarte en ello, también ofrece cierta protección legal el hecho de publicar tu trabajo en alguna parte, ya que entonces queda constancia pública de que tu nombre está relacionado con él (a menos que alguien haya hecho lo mismo con el mismo trabajo en un fecha anterior, claro).

En cuanto a tus imágenes, la catapulta es estupenda. Todo en ella es perfecto excepto la textura de la madera, que no parece estar a la altura del resto del conjunto. Pasando a piezas, el conjunto es visualmente atractivo pero podrías haber mejorado bastante la forma de las piezas de ajedrez y en cuanto a la textura azul... (¡ugh!).

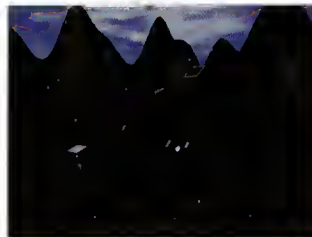
Y en cuanto a ti, Juan Ramon, la Sor del cañón no logra demasiado bien la forma esperada en un cañón medieval (más bien parece futurista). En cuanto a Sasino, ya conozco las dificultades de crear una figura antropomórfica desde cero... Quizá hubieras debido recubrirlo totalmente con una armadura de estilo medieval. Y... ¿a qué te refieres con eso de las rotaciones de Csg?

Isaac García Martínez, nos envía su primera imagen con 3D Studio. La imagen tiene un estilo naif bastante simpático que aún podría haber quedado mejor aplicando un bitmap "planeta tierra" sobre el mundo. Isaac también menciona una catapulta en fase de desarrollo para la inminente batallita. ¡Date prisa! ¡Si la batalla se celebrara en el próximo número te quedarías fuera! ¿Que por qué no regalamos Ipas de 3D Studio? Veras, los Ipas son creados por gente que aspira a vivir de su trabajo con las ventas. Solamente pueden regalarse versiones free o share de los mismos. En cuanto a los precios de los Ipas varían enormemente. Puedes encontrar un Ipa por 10000 o por 300000 pesetas. No tenemos por ahora una lista -ni siquiera medianamente completa- de distribuidores de Ipas.





El Foro del Lector

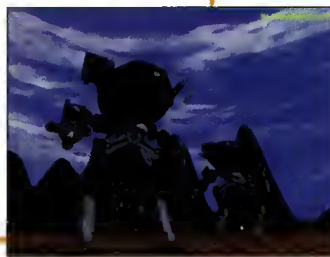


Julián Hierro García desea participar en la tan anunciada batallita virtual de orcos y humanos y envía para ello una catapulta que formara parte del bando orco. El modelo basado en el lanzador de rocas orco del mundo de Warhammer ha sido creado con Moray y es sencillo pero

muy resultón.

Más complejo es el Mecha, también hecho con Moray, enviado por Julián. La forma general y las articulaciones del "Mecha" son excelentes, aunque el resultado aún podría haber mejorado un poco con un cuerpo central de apariencia algo más compleja. En cuanto a tu sugerencia acerca de un artículo sobre animación con POV... Bien, es cierto que tendría que haberse hablado de ello hace tiempo. Intentaremos remediar esta situación en breve. Por último, Julián lanza una propuesta interesante (leed la carta en el foro): hacer una colección de modelos tipo Warhammer y otra tipo Battletech para hacer batallitas. Esta claro que, si la gente se anima a ello, habría que publicar unas reglas mínimas sobre proporciones, orientación de los modelos, conversiones de formato, etc porque, claro está, cada uno emplearía una herramienta distinta.

¿Qué dicen los rendermaniacos? Por de pronto nuestro seguro servidor aporta el modelo de ejemplo de este mes.



Luis Pérez Monpeán hace su primera aparición en nuestro foro con dos imágenes creadas con POV. Una de ellas, la de la bola, es un experimento con las normales del objeto que ha dado un resultado bastante vistoso. Nos sumamos a tus felicitaciones a Julián Hierro García que, como podrás comprobar en el número de hoy, ha seguido mejorando con respecto a aquella primera imagen del número 3 de Rendermanía. ¡Ánimo! Tú también puedes mejorar. ¡Atención Nasser Afkir Torres! Luis te hace un par de comentarios en su carta, en el foro.

Luis Cardenas Salido nos ha enviado un magnífico AVI en el que pueden verse efectos de nieve, movimientos de burbujas dentro de líquidos, fuego, luces flare, etc. La creación de esta estupenda animación, que incluye una banda sonora, ha requerido el empleo de bastantes programas cuya lista podéis ver en la carta del foro. ¡Enhorabuena Luis!

NOTA SOBRE IMAGINE

Algunos lectores nos han enviado notas informando de problemas con el funcionamiento de Imagine. Resulta difícil responder a ellas ya que nosotros no hemos apreciado ningún problema, en todo caso recordamos que la versión distribuida en PCmanía funciona bajo MS Dos. Esta versión no se ejecutará desde Windows. Habréis de salir al Dos y -para quienes lo hayáis hecho así- el programa gestionará el mismo la memoria por lo que no debe haber otro gestor. Además, en la versión anterior de Imagine se recomendaba también liberar toda la memoria posible del primer megabyte (por ejemplo, eliminando programas residentes).

Luis España Martín nos envía un pequeño artículo en formato MS Word que reproducimos en el foro. En él, Luis explica en detalle el proceso seguido con 3D Studio para crear la efectista escena que podéis ver aquí: la iluminación, los Ipas utilizados, las texturas, la

cámara, etc. Enhorabuena por tu escena. No es frecuente ver por aquí escenas con "tratamiento atmosférico" hechas con 3D Studio.

